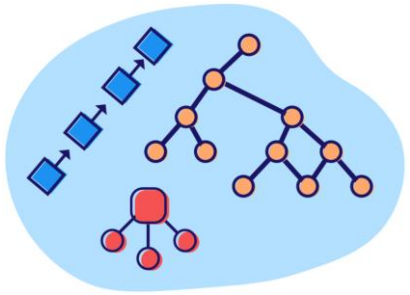
الجامعة العربية الخاصة للعلوم والتكنولوجيا
كلية الهندسة المعلوماتية
قسم تقانة المعلومات

Algorithms & Data Structures (2)

Lecture-2: Graph
(البيان)



Dr. Asmaa Shaar

فهرس المحاضرة

- التجوال عبر البيان:
  - باستخدام خوارزمية البحث بالعرض أولاً (BFS).
  - باستخدام خوارزمية البحث بالعمق أولاً (DFS).
- تطبيقات خوارزميات التجوال
- الفرز الطوبولوجي (Topological Sort)

التجوال عبر البيان

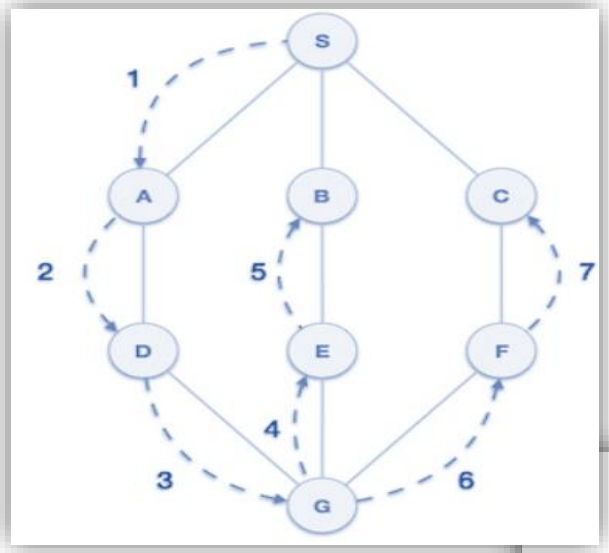
- عبارة عن تقنية لزيارة كل عقدة في البيان **بشكل منهجي** ومعالجتها لمرة واحدة بالضبط.
  - اتخاذ القرار حول **ترتيب زيارة عقد البيان المُتصلة ببعضها البّعض** اثناء عملية معالجة ما دون الوقوع في **مسار حلقي** (looping path).

- يتم استخدامها:
  - للبحث عن **عقدة ما** ضمن البيان.
  - ايجاد **مسار** من عقدة لاخرى.
  - تحديد فيما اذا كان البيان **متصل** (connected) ام لا
  - ايجاد **اشجار الامتداد** (spanning tree)

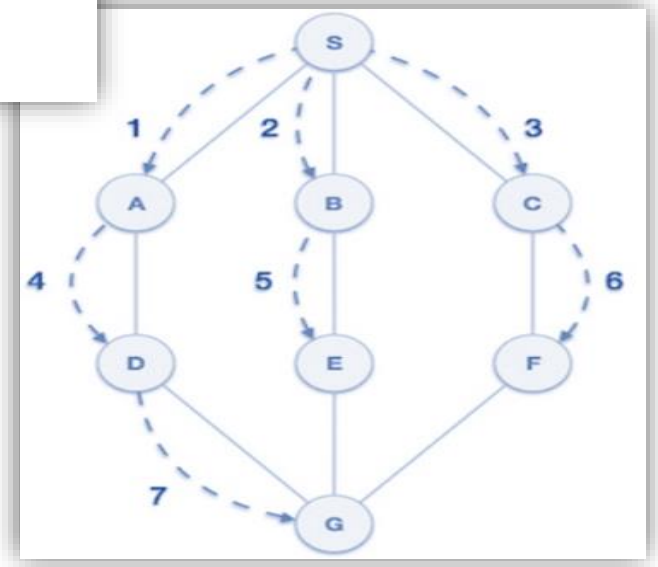
- يتم تنفيذ التجوال عبر البيان باستخدام:
  - BFS: خوارزمية البحث بالعرض أولاً (breadth first search).
  - DFS: خوارزمية البحث بالعمق أولاً (depth first search).

التجوال عبر البيان

- **ملاحظة:** لا تختلف طرق التجوال والإجرائيات المستخدمة في **البيان الموجه** عن تلك الإجرائيات المطبقة في البحث عبر **بيان غير موجه** والفرق الوحيد هو مراعاة اتجاه الأضلاع.

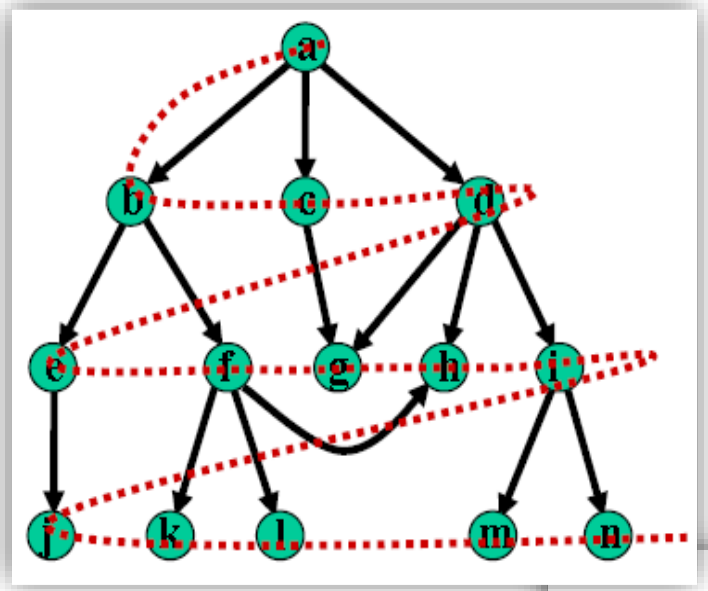


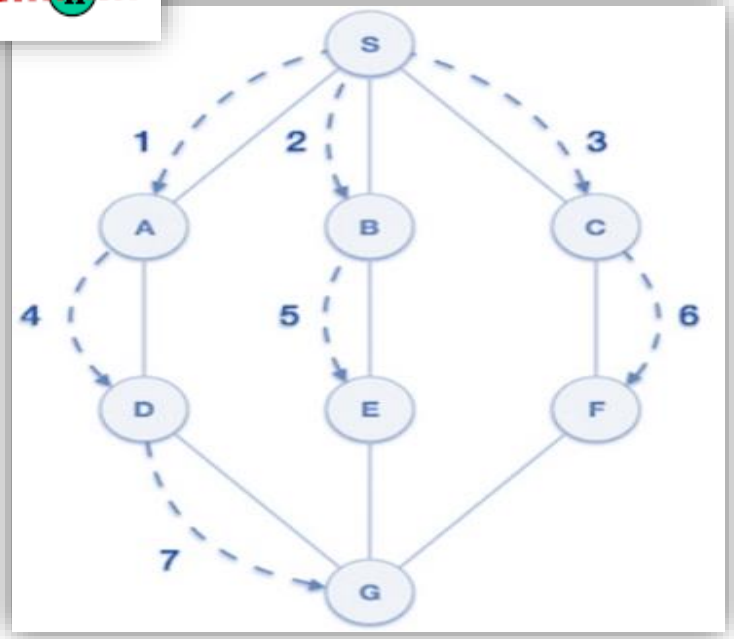
- **ملاحظة:** التجوال باستخدامBFS و DFSللبيان ليس وحيداً فهو يعتمد على كل من
  - عقدة البداية،
  - وعلى ترتيب زيارة الرؤوس المجاورة لكل عقدة.



التجوال عبر البيان
(Breadth First Search)

البحث بالعرض أولاً





- مثال: بفرض G=<V,E> بيان ولتكن $s \in V(G)$ **عقدة البداية**.
  - نبدأ التجوال من عقدة البداية المختارة s ثم العقد المجاورة لها(العقد التي ترتبط مباشرة بعقدة البداية).
  - يتم الانتقال نحو العقد المجاورة في المستوى التالي و **التي لم تتم زيارتها بعد.**
  - وهكذا.....

البحث بالعرض أولاً

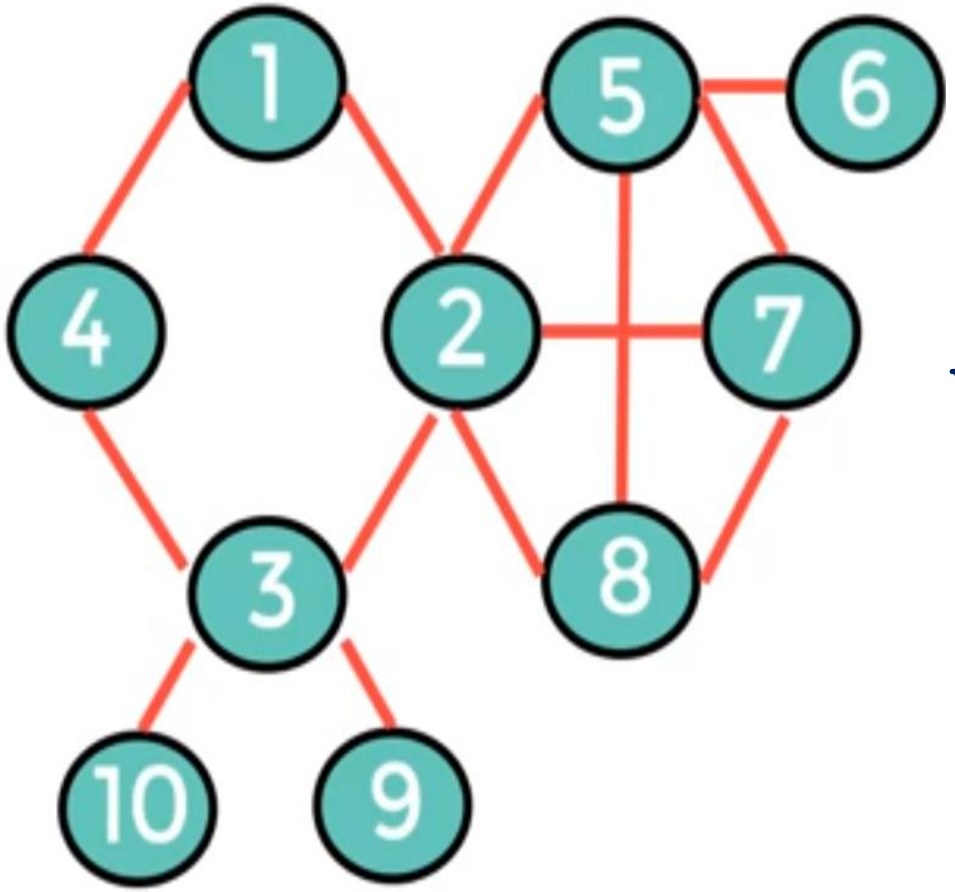
- نوضح خوارزمية البحث بالعرض أولاً باستخدام pseudocode على الشكل التالي :

```
void BFS(int s)                                  //s دليل عقدة البداية،نعتبر أن البيان مخزن في مصفوفة بوليانية
{ Queue q;                                       //q عبارة عن رتل، حيث تعتمد الخوارزمية على مبدأ الرتل (FIFO) //
  char visited[n];                               //visited عبارة عن مصفوفة لمعرفة العقد التي تمت زيارتها//
  for (int i=0;i<n;i++)                          //n عبارة ع عدد العقد في البيان//
     visited[i]='n';                             //في البداية، جميع العقد لم تتم زيارتها//

  visited[s]='y';                                //يتم زيارة عقدة البداية ونضعها في الرتل//
  q.enqueue(s);

while (q.isEmpty() == no)                        //طالما هناك عقد في الرتل (الرتل غير فارغ)، نكرر ما يلي://
{     v=q.dequeue();                             //1.يتم اخراج عقدة من بداية الرتل ولتكن v//
      for (all vertices w adjacent to v)         //2.من أجل كل عقدة w مجاورة لـ v ولم تتم زيارتها بعد //
           if (visited[w]=='n')
              {q.enqueue(w);                     //نضع w في الرتل ونشير إلى انه تمت زيارتها//
                visited[w]='y';}
}}
```

مثال باستخدام BFS



Q

	1	2	3	4	5	6	7	8	9	10
visited	n	n	n	n	n	n	n	n	n	n

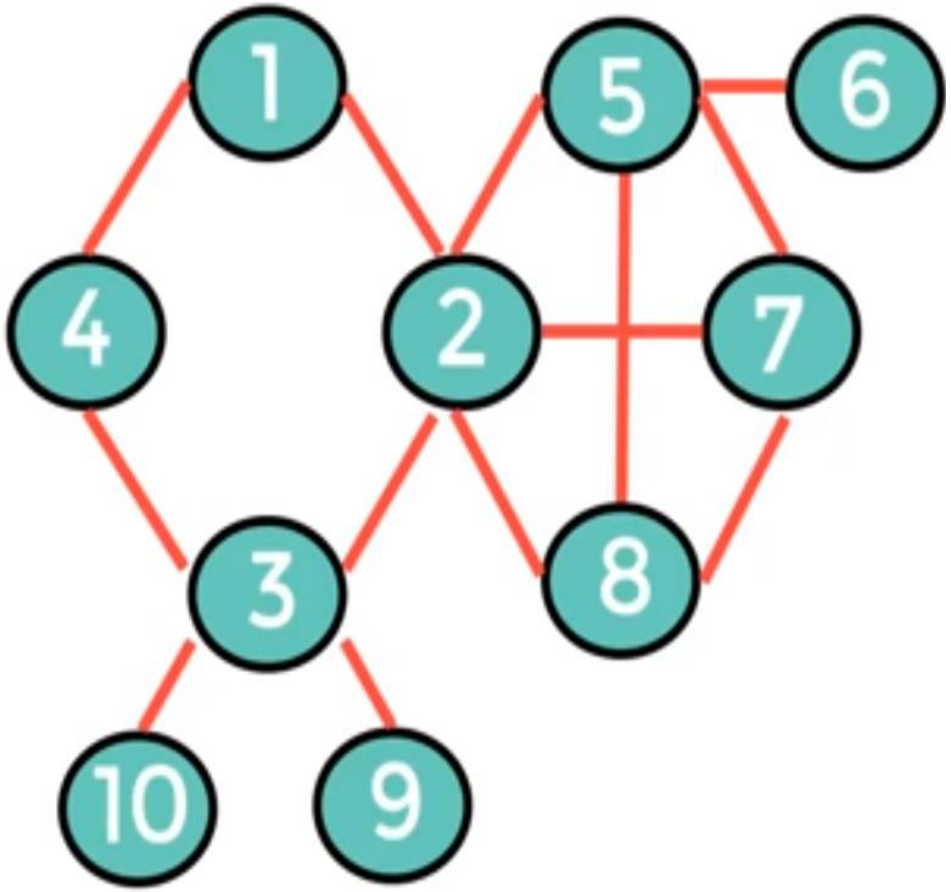
Q 1

	1	2	3	4	5	6	7	8	9	10
visited	y	n	n	n	n	n	n	n	n	n

Start node: 1

ملاحظة: يمكن إدراج الرؤوس المجاورة في الرتل بأي ترتيب؛ ولكن للحصول على **تجوال وحيد** نضعهم في ترتيب أبجدي أو من اليسار إلى اليمين.

مثال باستخدام BFS



BFS: 1

Q

visited

1	2	3	4	5	6	7	8	9	10
y	y	n	y	n	n	n	n	n	n

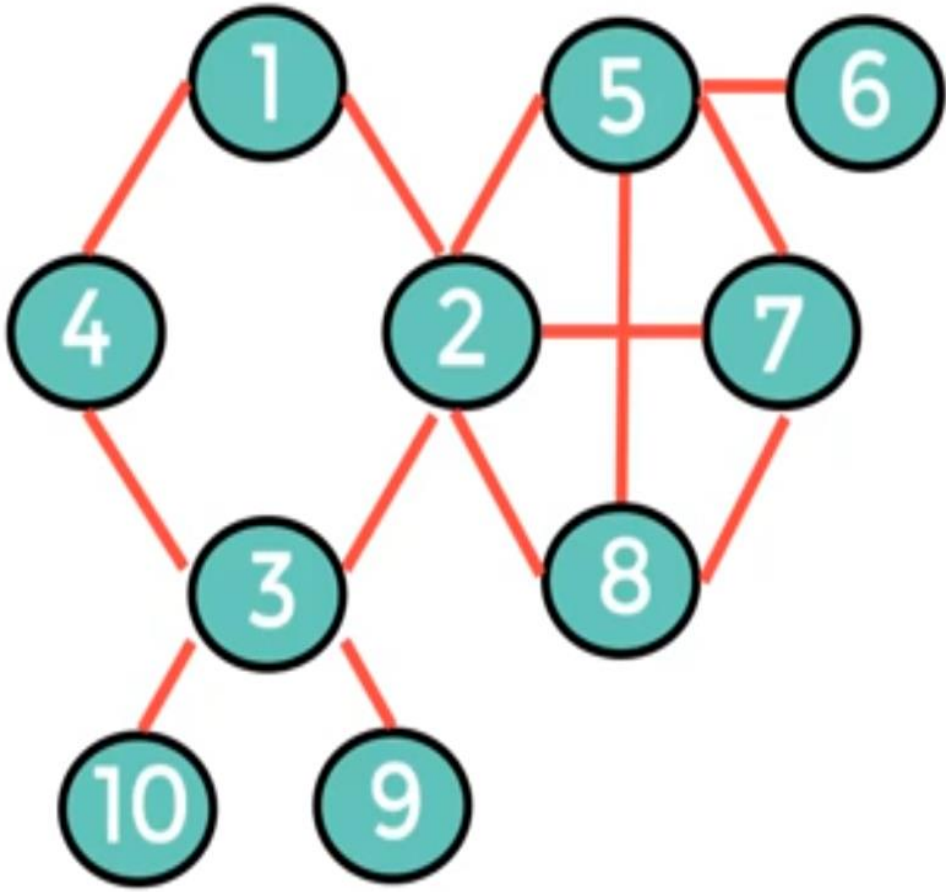
BFS: 1 4

Q

visited

1	2	3	4	5	6	7	8	9	10
y	y	y	y	n	n	n	n	n	n

مثال باستخدام BFS



BFS: 1 4 2

Q

visited

1	2	3	4	5	6	7	8	9	10
y	y	y	y	y	n	y	y	n	n

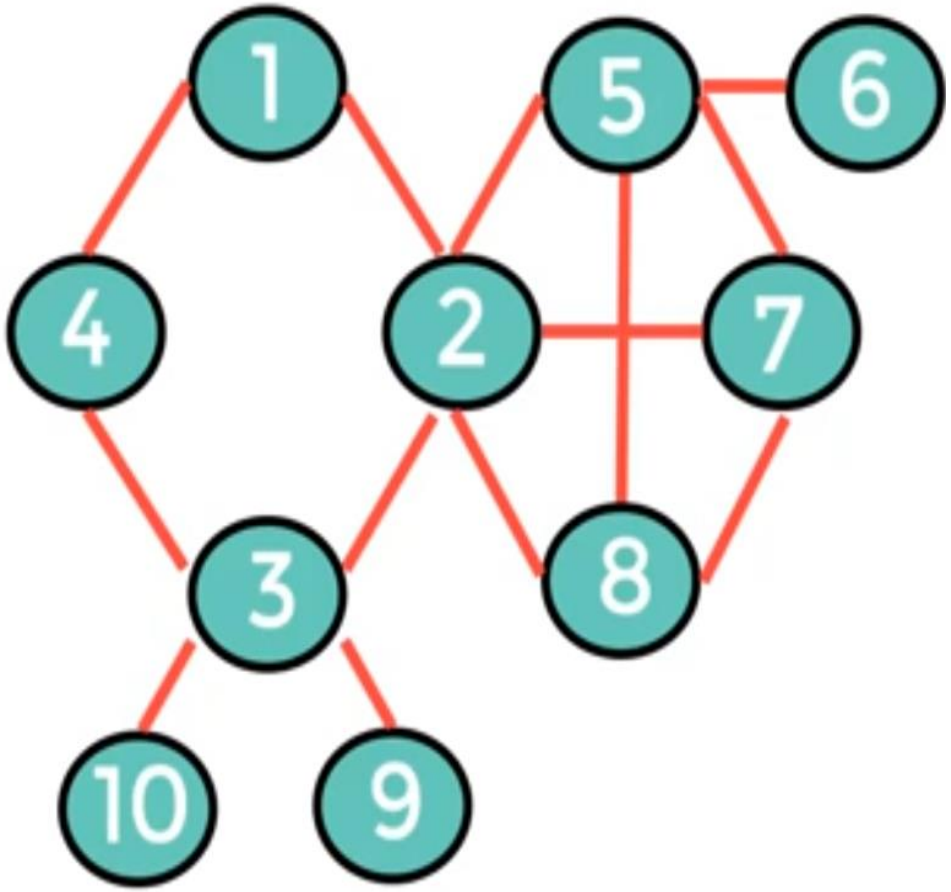
BFS: 1 4 2 3

Q

visited

1	2	3	4	5	6	7	8	9	10
y	y	y	y	y	n	y	y	y	y

مثال باستخدام BFS



BFS: 1 4 2 3 5

Q

visited

1	2	3	4	5	6	7	8	9	10
y	y	y	y	y	y	y	y	y	y

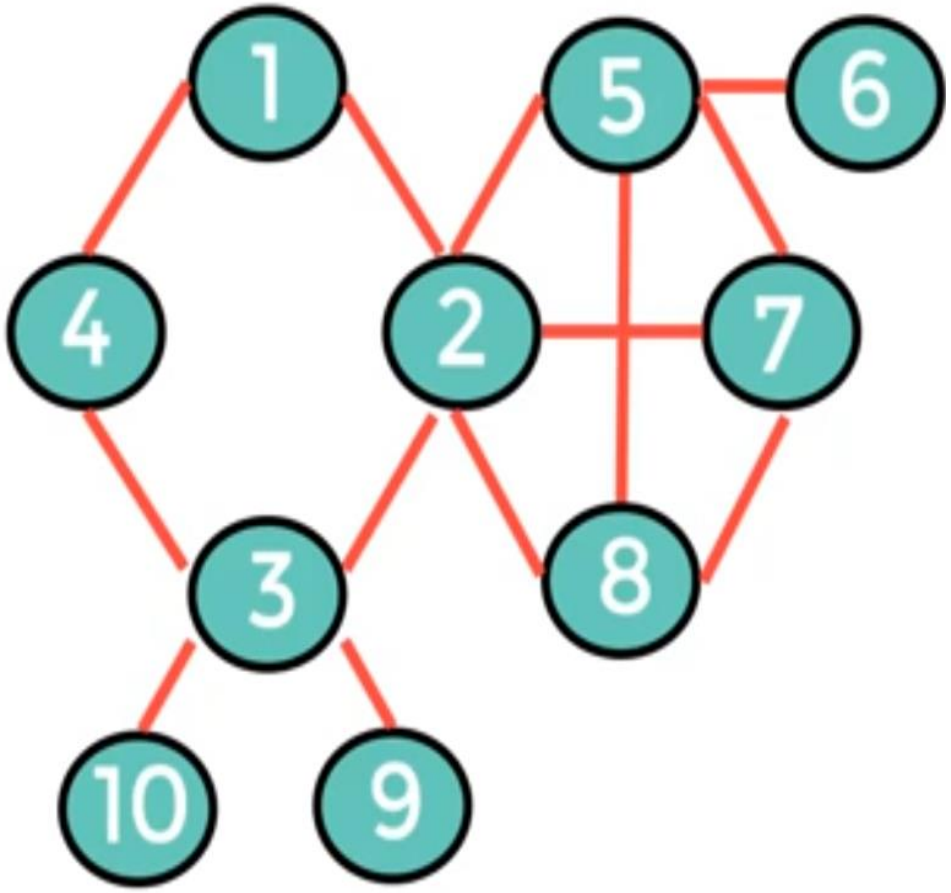
BFS: 1 4 2 3 5 8

Q

BFS: 1 4 2 3 5 8 7

Q

مثال باستخدام BFS



BFS: 1 4 2 3 5 8 7 10

Q 1 4 2 3 5 8 7 10 9 6

BFS: 1 4 2 3 5 8 7 10 9

Q 1 4 2 3 5 8 7 10 9 6

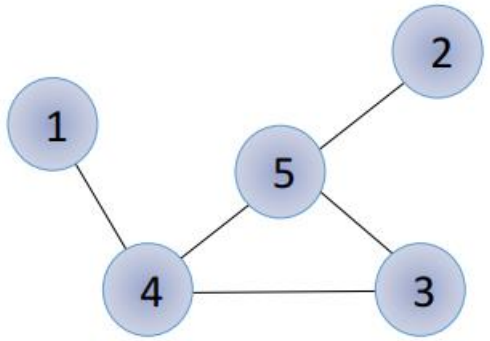
BFS: 1 4 2 3 5 8 7 10 9 6

Q 1 4 2 3 5 8 7 10 9 6

تطبيقات خوارزمية BFS

- يمكن **لمحركات البحث** إنشاء الفهارس الخاصة بها عن طريق استخدام خوارزمية BFS التي يبدأ تطبيقها من صفحة ويب مصدر ما ومن ثم تتم زيارة جميع صفحات الويب التي تشير إليها روابط صفحة المصدر هذه وهكذا...
- أنظمة الملاحة الجغرافية (GPS Navigation systems): يمكن أن تساعد خوارزمية BFS في العثور على جميع **المواقع المجاورة** لموقع هدف.
- شبكات الند للند (P2P): يمكن استخدام خوارزمية BFS لإيجاد العقد **الأقرب أو المجاورة** في شبكات الند للند والذي بدوره سيؤدي لإيجاد **البيانات المطلوبة بشكل أسرع**.
- Network Broadcasting.
- هي طريقة **لايجاد أقصر** طريق في **بيان غير موزون** لأن الوصول للعقد يتم بمسح أقل عدد للروابط (يتم استخدام مصفوفة لتخزين أقصر طريق للعقدة عند اول وصول اليه).
- .....

تمارين

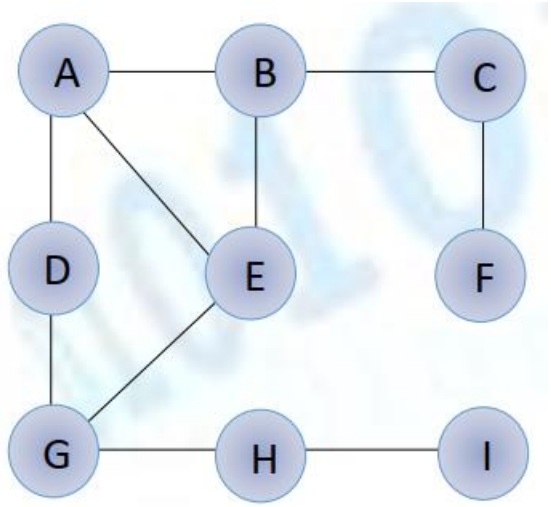


مثال:

نقطة البداية هي العقدة رقم 5

فتكون أحد طرق زيارة الغراف المجاور بالشكل التالى باتباع خوارزمية BFS :

52341



مثال:

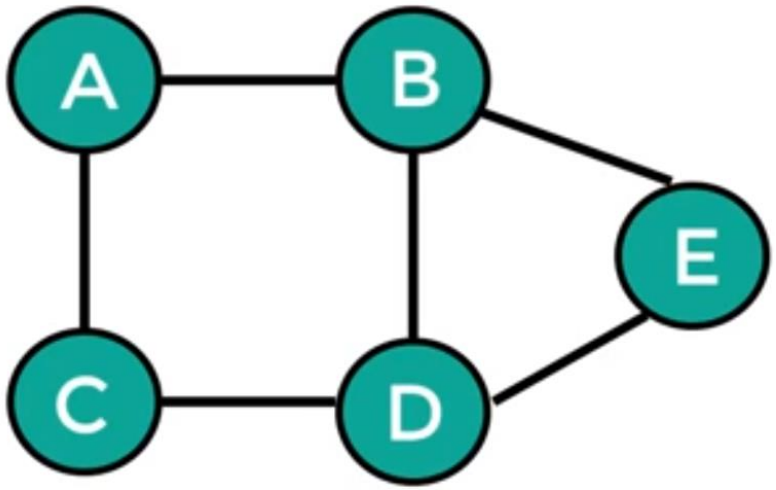
نقطة البداية هي العقدة F

فتكون أحد طرق زيارة الغراف المجاور بالشكل التالي باتباع خوارزمية BFS :

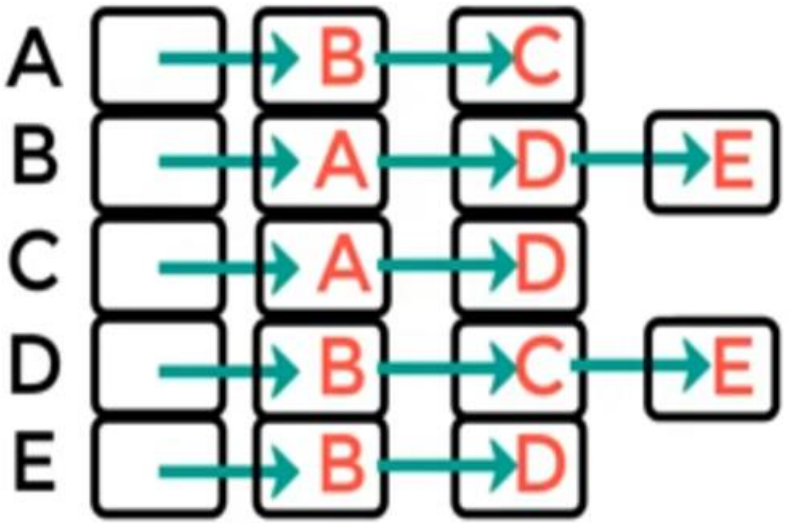
FCBAEDGHI

تحليل الـ BFS

- من أجل البيان G الذي من يملك |V(G)|=n, |E(G)| =e. يعتمد تعقيد الـ BFS فيه على طريقة التخزين:
  - إذا كان البيان ممثلاً بمصفوفة التجاور يكون التعقيد هو $O(n^2)$.
    - ✓ تحديد الرؤوس المتجاورة مع الرأس v يتطلب زمناً قدره O(n) و بما أنه يتم زيارة كل الرؤوس على الأكثر لذلك يكون زمن التنفيذ الكلي
  - إذا كان البيان ممثلاً بقوائم التجاور يكون التعقيد هو $O(n+e)$ وذلك لاننا سنمر في الخوارزمية على e ضلع على الاكثر و يضاف لها كلفة زيارة كل عقدة من العقد.
    - ✓ الكلفة الكلية للحلقة while هو $d_1+d_2+\ldots+d_n= O(e)$ حيث $d_i=degree(i)$.

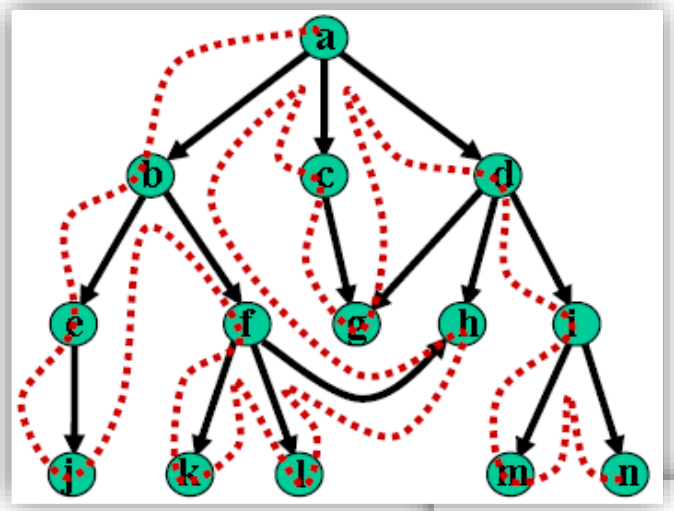


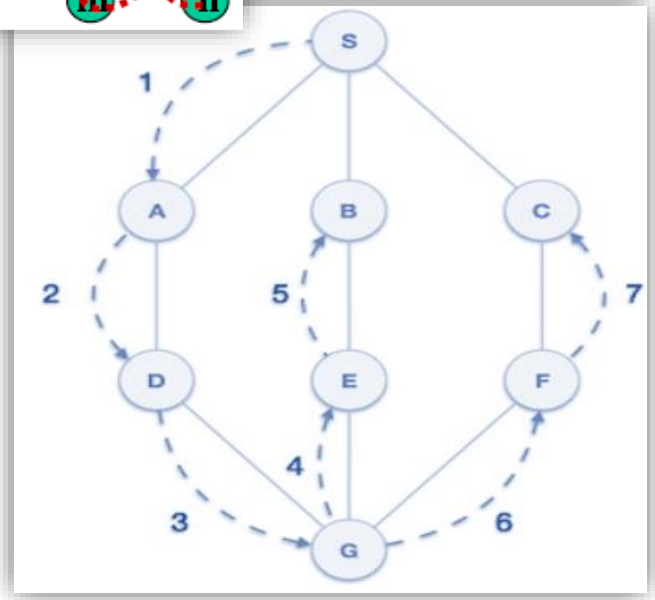
	A	B	C	D	E
A	0	1	1	0	0
B	1	0	0	1	1
C	1	0	0	1	0
D	0	1	1	0	1
E	0	1	0	1	0



التجوال عبر البيان
(Depth First Search)

خوارزمية البحث بالعمق أولا





- مثال: بفرض G=<V,E> بيان ولتكن $s \in V(G)$ عقدة البداية.
  - نبدأ التجوال من عقدة البداية المختارة s، ثم نختار عقدة مجاورة لها w **لم تتم زيارتها بعد**، ونتابع عند هذه العقدة **بطريقة عودية**.
  - عندما نصل إلى عقدة ليس لها عقد مجاورة لم تتم زيارتها، **نعود** إلى آخر عقدة تمت زيارتها ولها عقد مجاورة لم تتم زيارتها ونتابع عند هذه العقدة **بطريقة عودية**.
  - تنتهي الخوارزمية عند زيارة جميع عقد البيان.
  - يمكن تحقيق هذا التجوال باستخدام الاسلوب العودي أو التكراري.

الخوارزمية العودية للبحث بالعمق أولا

- نوضح **الخوارزمية العودية** للبحث بالعمق أولاً باستخدام pseudocode على الشكل التالي:

```
char visited[n];
for (int i=0;i<n;i++)
     visited[i]='n';
```

visited عبارة عن مصفوفة لمعرفة العقد التي تمت زيارتها//
n عبارة ع عدد العقد في البيان//
في البداية، جميع العقد لم تتم زيارتها//

```
void DFS(int s) {
  visited[s]='y';
  for (each vertex w adjacent to s)
      if (visited[w]=='n')
          DFS(w);
}
```

S = دليل العقدة المراد زيارتها، بحيث عند أول استدعاء يكون دليل عقدة البداية

نختار عقدة w مجاورة لـ S لم تتم زيارتها بعد//
نتابع عند هذه العقدة بطريقة عودية//

الخوارزمية التكرارية للبحث بالعمق أولا

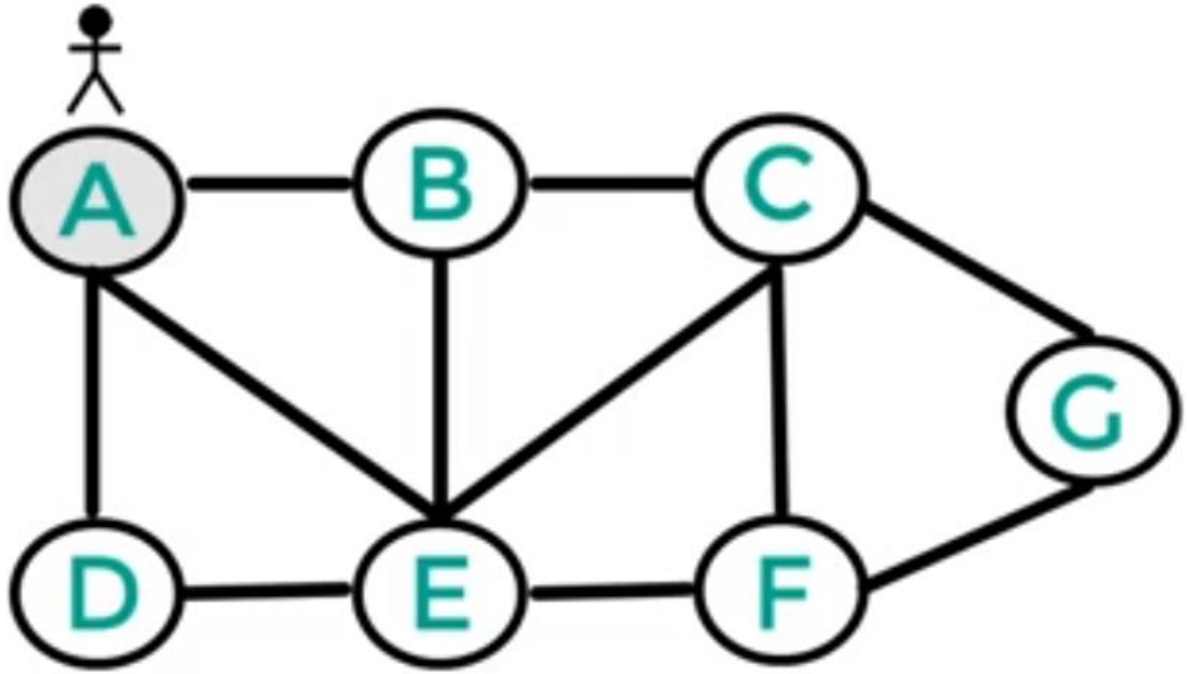
- يتم تحويل الدالة العودية لـ dfs إلى دالة تكرارية **باستخدام المكدس** وفقاً للخطوات التالية:

1. نقوم بتعريف **مكدس** حجمه يساوي عدد العقد في البيان.
2. نختار أي عقدة من البيان **كنقطة بداية** للتجوال ونقوم بزيارة تلك العقدة ونضعها في المكدس.
3. نقوم بزيارة إحدى العقد المجاورة للعقدة الموجودة في قمة المكدس والتي **لم تتم زيارتها** بعد و نضعها في المكدس.
4. نكرر الخطوة 3، حتى **لا يبقى أي عقدة مجاورة** يمكن زيارتها من العقدة الموجودة في قمة المكدس.
5. نقوم **بالتراجع وإخراج** عقدة من المكدس.
6. نكرر الخطوات 3،4،5 حتى يصبح المكدس **فارغ**

```
push the starting vertex onto the stack
while(stack is not empty) {
    pop a vertex off the stack, call it v
    if v is not already visited, visit it
    push vertices adjacent to v, not visited, onto the stack
                        }
```

مثال باستخدام DFS

1. نقوم بتعريف **مكدس** حجمه يساوي عدد العقد في البيان.
2. نختار أي عقدة من البيان كنقطة بداية للتجوال ونقوم بزيارة تلك العقدة ونضعها في المكدس.

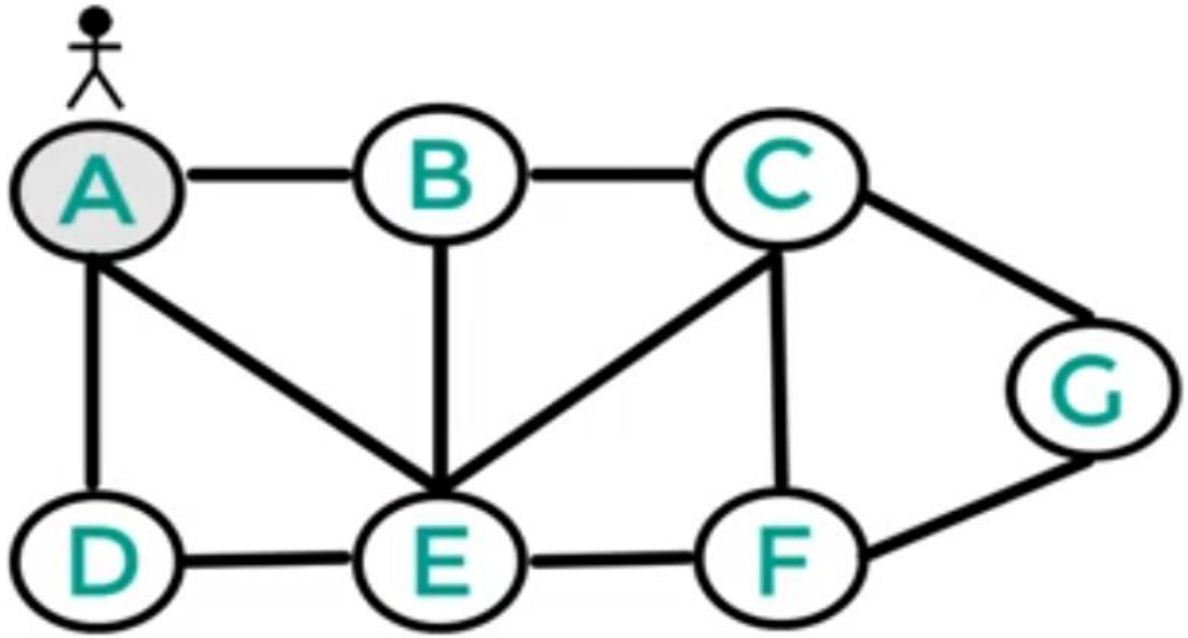


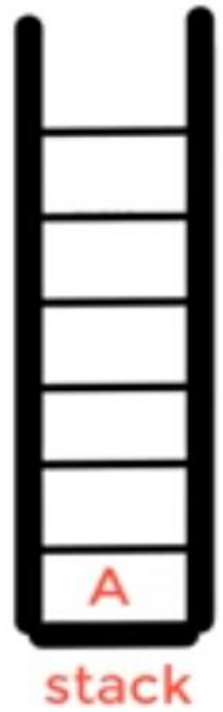
visited

A	B	C	D	E	F	G
y	n	n	n	n	n	n

مثال باستخدام DFS

1. نقوم بتعريف **مكدس** حجمه يساوي عدد العقد في البيان.
2. نختار أي عقدة من البيان كنقطة بداية للتجوال ونقوم بزيارة تلك العقدة ونضعها في المكدس.
3. نقوم بزيارة إحدى العقد المجاورة للعقدة الموجودة في قمة المكدس والتي لم تتم زيارتها بعد و نضعها في المكدس.





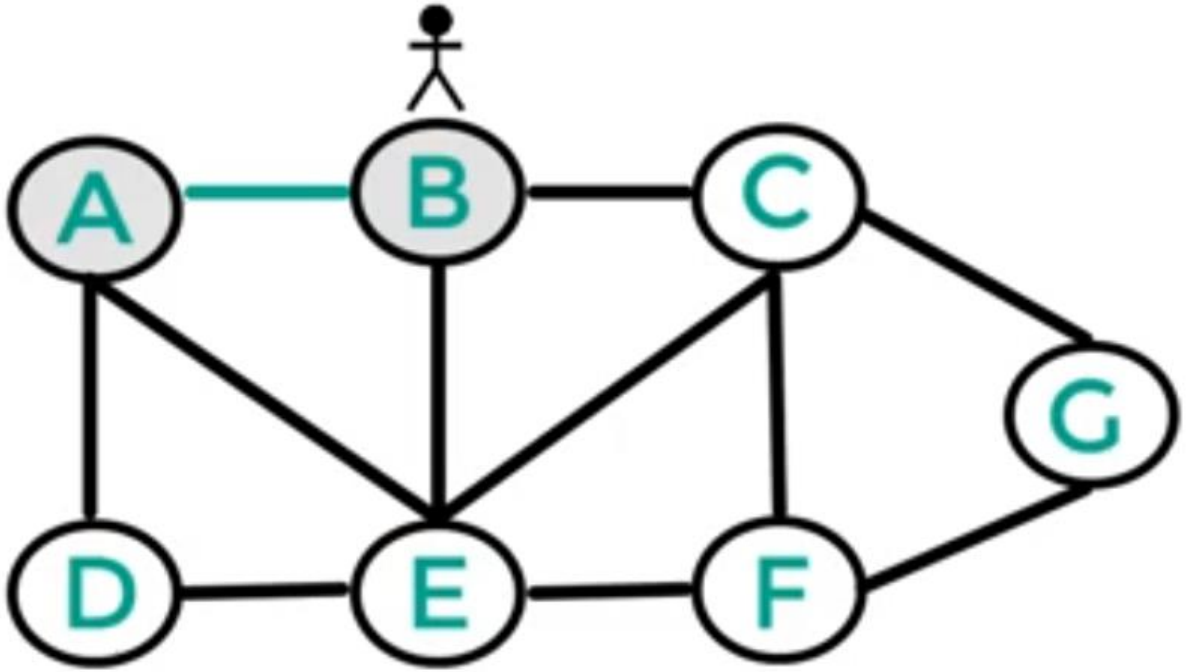
visited

A	B	C	D	E	F	G
y	n	n	n	n	n	n

ملاحظة: يمكن إدراج الرؤوس المجاورة في المكدس بأي ترتيب؛ ولكن للحصول على **تجوال وحيد** نضعهم في ترتيب أبجدي.

مثال باستخدام DFS

1. نقوم بتعريف **مكدس** حجمه يساوي عدد العقد في البيان.
2. نختار أي عقدة من البيان كنقطة بداية للتجوال ونقوم بزيارة تلك العقدة ونضعها في المكدس.
3. نقوم بزيارة إحدى العقد المجاورة للعقدة الموجودة في قمة المكدس والتي لم تتم زيارتها بعد و نضعها في المكدس.

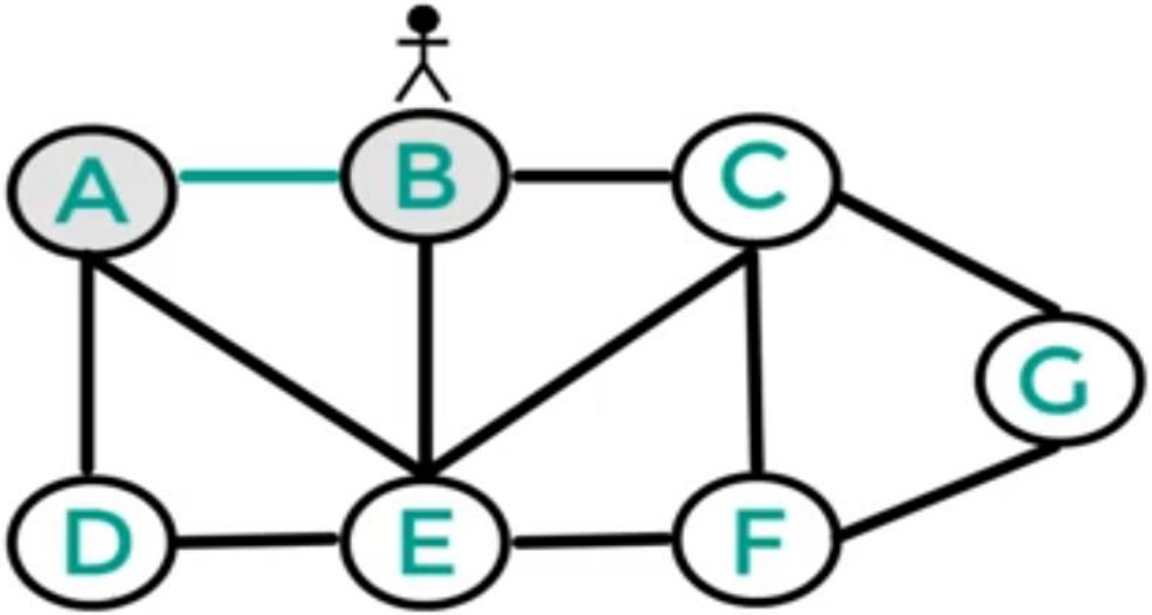


visited

A	B	C	D	E	F	G
y	y	n	n	n	n	n

مثال باستخدام DFS

1. نقوم بتعريف **مكدس** حجمه يساوي عدد العقد في البيان.
2. نختار أي عقدة من البيان كنقطة بداية للتجوال ونقوم بزيارة تلك العقدة ونضعها في المكدس.
3. نقوم بزيارة إحدى العقد المجاورة للعقدة الموجودة في قمة المكدس والتي لم تتم زيارتها بعد و نضعها في المكدس.
4. نكرر الخطوة 3 ، حتى لا يبقى أي عقدة مجاورة يمكن زيارتها من العقدة الموجودة في قمة المكدس.

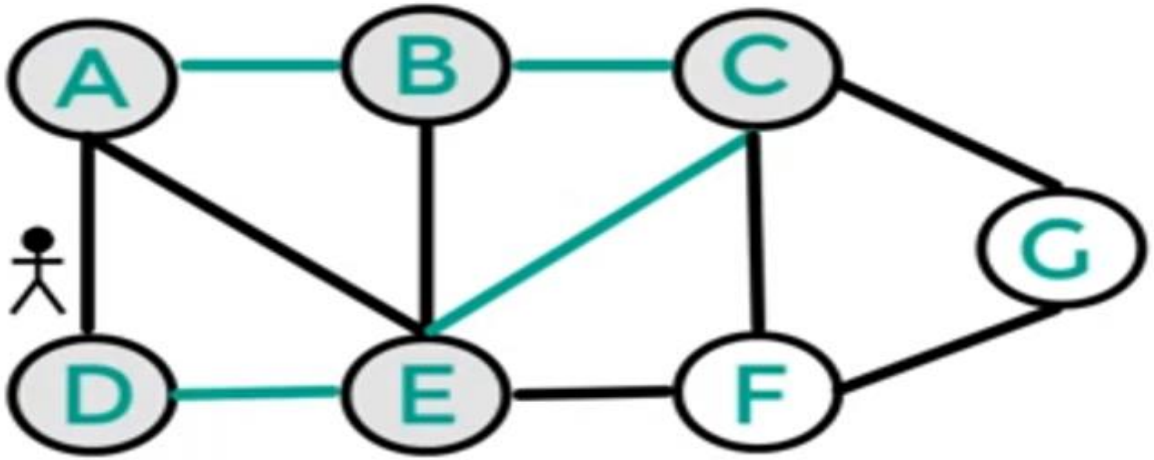


visited

A	B	C	D	E	F	G
y	y	n	n	n	n	n

مثال باستخدام DFS

1. نقوم بتعريف **مكدس** حجمه يساوي عدد العقد في البيان.
2. نختار أي عقدة من البيان كنقطة بداية للتجوال ونقوم بزيارة تلك العقدة ونضعها في المكدس.
3. نقوم بزيارة إحدى العقد المجاورة للعقدة الموجودة في قمة المكدس والتي لم تتم زيارتها بعد و نضعها في المكدس.
4. نكرر الخطوة 3، حتى لا يبقى أي عقدة مجاورة يمكن زيارتها من العقدة الموجودة في قمة المكدس.
5. نقوم بالتراجع وإخراج عقدة من المكدس.

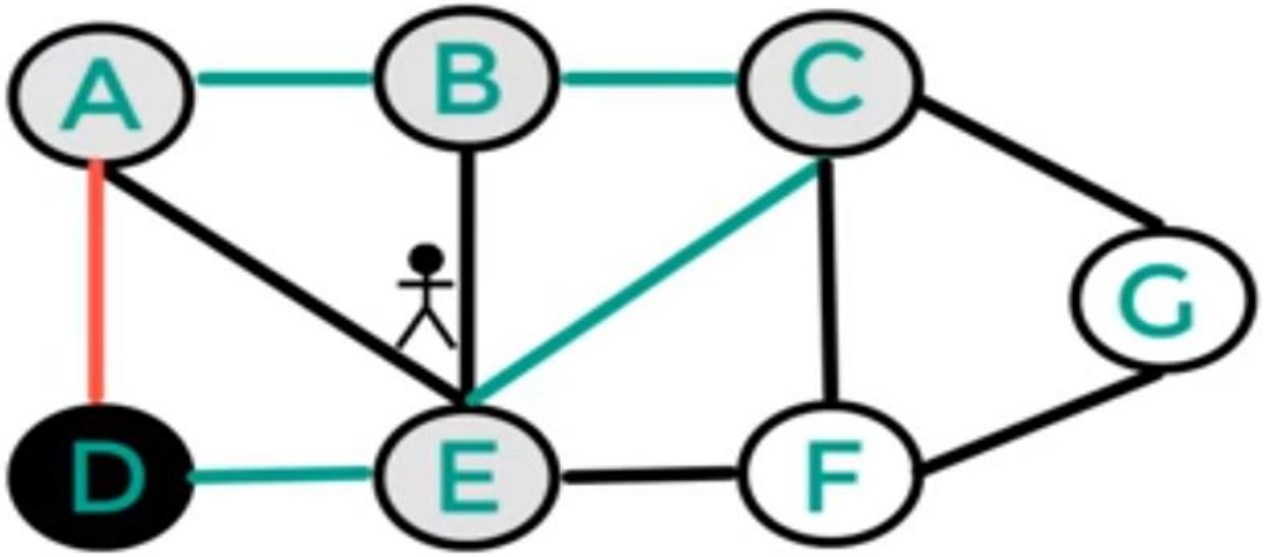


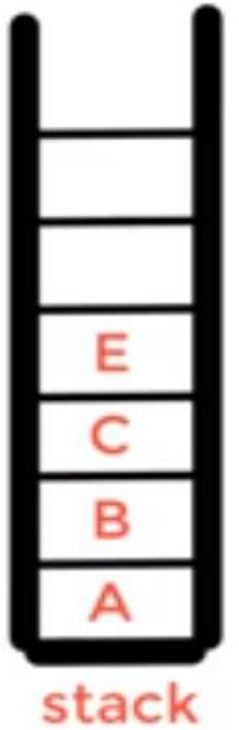
visited

A	B	C	D	E	F	G
y	y	y	y	y	n	n

مثال باستخدام DFS

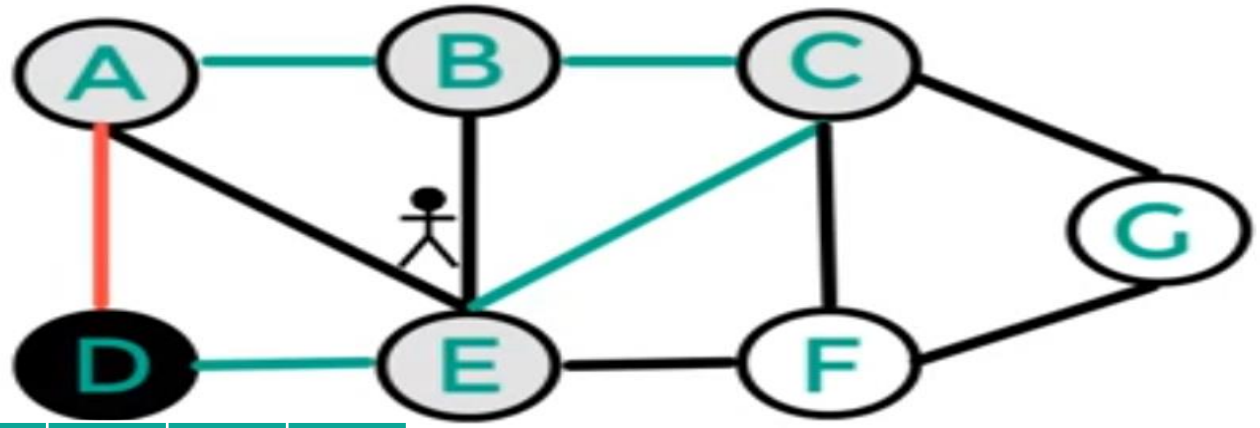
1. نقوم بتعريف **مكدس** حجمه يساوي عدد العقد في البيان.
2. نختار أي عقدة من البيان كنقطة بداية للتجوال ونقوم بزيارة تلك العقدة ونضعها في المكدس.
3. نقوم بزيارة إحدى العقد المجاورة للعقدة الموجودة في قمة المكدس والتي لم تتم زيارتها بعد و نضعها في المكدس.
4. نكرر الخطوة 3، حتى لا يبقى أي عقدة مجاورة يمكن زيارتها من العقدة الموجودة في قمة المكدس.
5. نقوم بالتراجع وإخراج عقدة من المكدس.

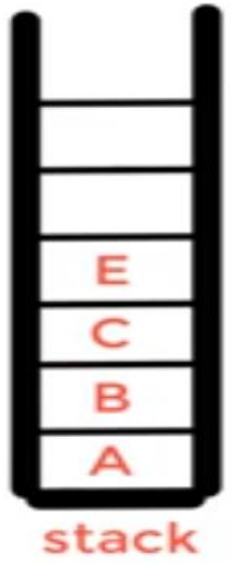




مثال باستخدام DFS

1. نقوم بتعريف **مكدس** حجمه يساوي عدد العقد في البيان.
2. نختار أي عقدة من البيان كنقطة بداية للتجوال ونقوم بزيارة تلك العقدة ونضعها في المكدس.
3. نقوم بزيارة إحدى العقد المجاورة للعقدة الموجودة في قمة المكدس والتي لم تتم زيارتها بعد و نضعها في المكدس.
4. نكرر الخطوة 3، حتى لا يبقى أي عقدة مجاورة يمكن زيارتها من العقدة الموجودة في قمة المكدس.
5. نقوم بالتراجع وإخراج عقدة من المكدس.
6. نكرر الخطوات 3،4،5 حتى يصبح المكدس فارغ



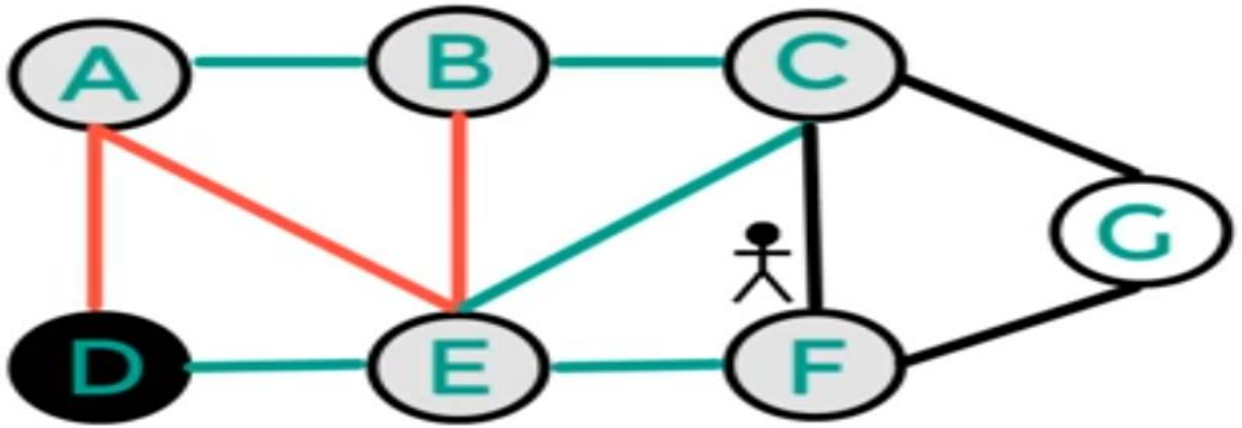


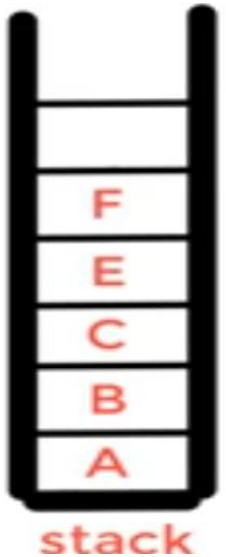
visited

A	B	C	D	E	F	G
y	y	y	y	y	n	n

مثال باستخدام DFS

1. نقوم بتعريف **مكدس** حجمه يساوي عدد العقد في البيان.
2. نختار أي عقدة من البيان كنقطة بداية للتجوال ونقوم بزيارة تلك العقدة ونضعها في المكدس.
3. نقوم بزيارة إحدى العقد المجاورة للعقدة الموجودة في قمة المكدس والتي لم تتم زيارتها بعد ونضعها في المكدس.
4. نكرر الخطوة 3، حتى لا يبقى أي عقدة جديدة يمكن مجاورة من العقدة الموجودة في قمة المكدس.
5. نقوم بالتراجع وإخراج عقدة من المكدس.
6. نكرر الخطوات 3،4،5 حتى يصبح المكدس فارغ

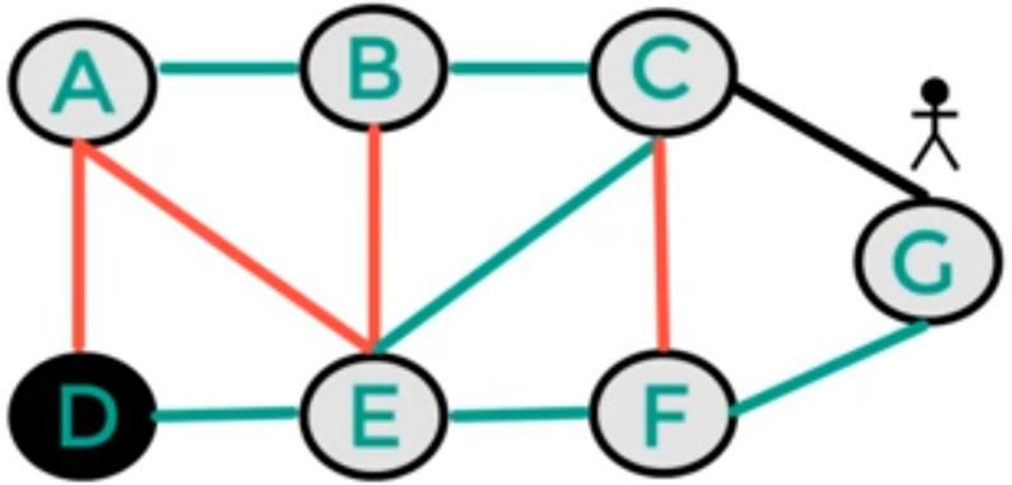


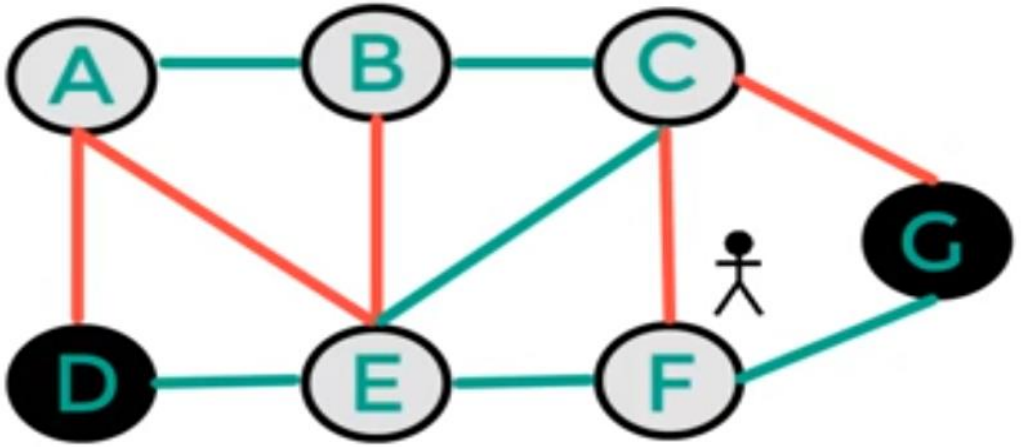


visited

A	B	C	D	E	F	G
y	y	y	y	y	y	n

مثال باستخدام DFS

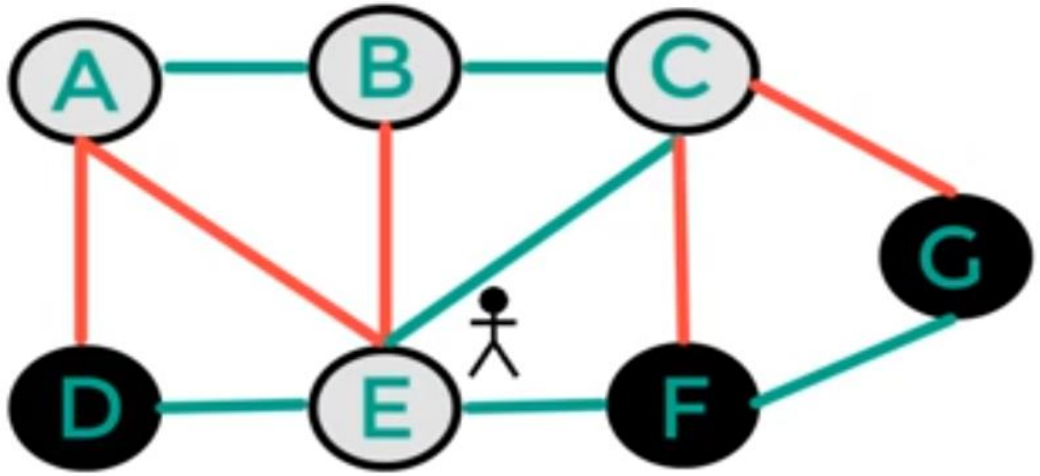


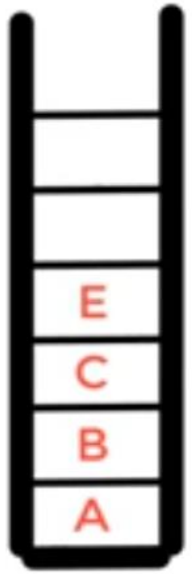


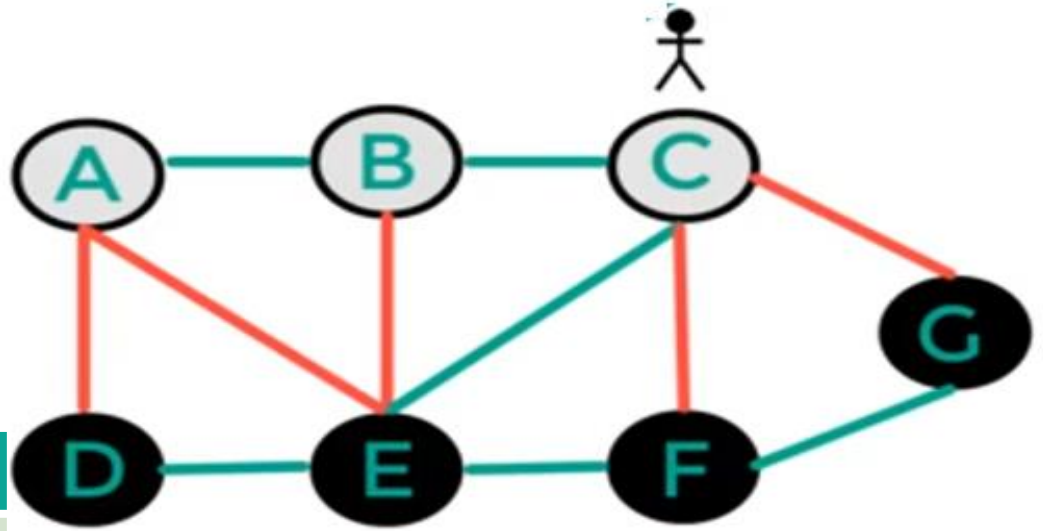
visited

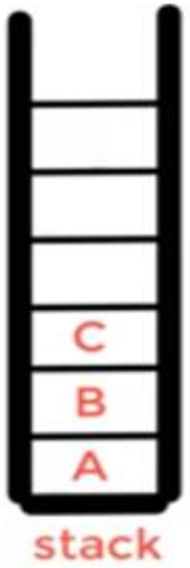
A	B	C	D	E	F	G
y	y	y	y	y	y	y

مثال باستخدام DFS





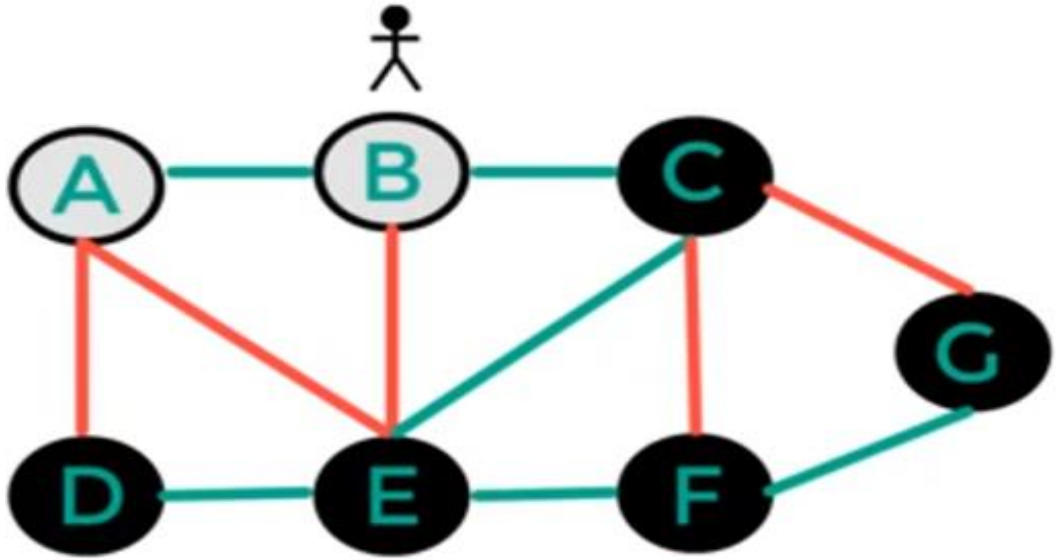


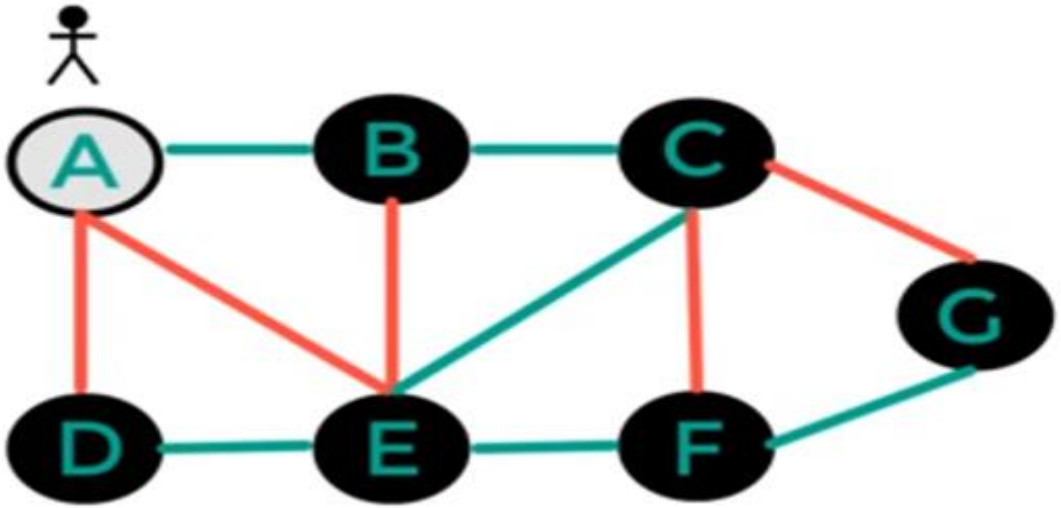


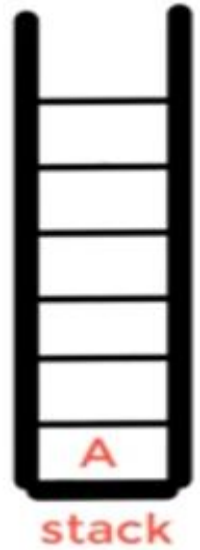
visited

A	B	C	D	E	F	G
y	y	y	y	y	y	y

مثال باستخدام DFS



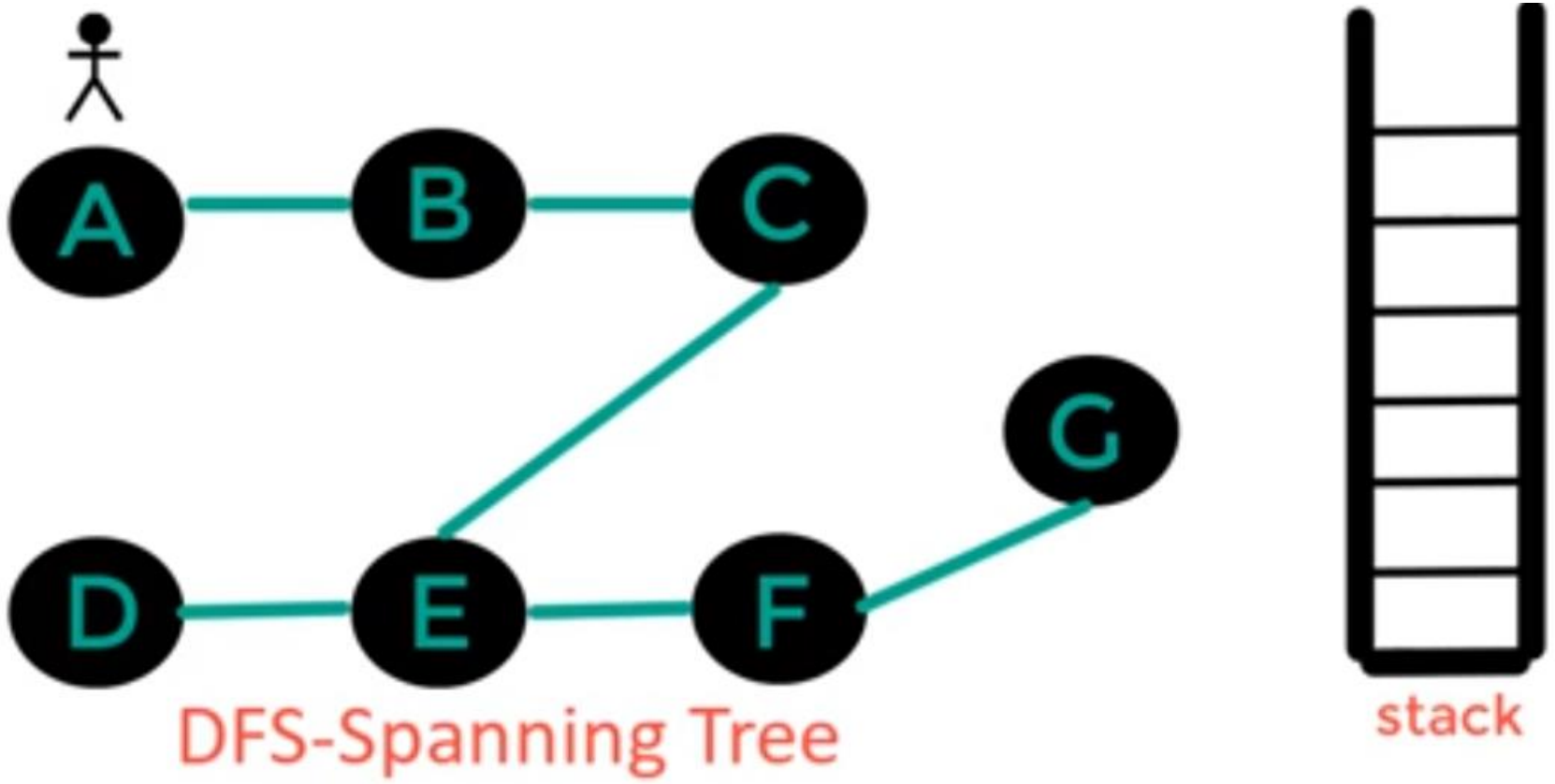




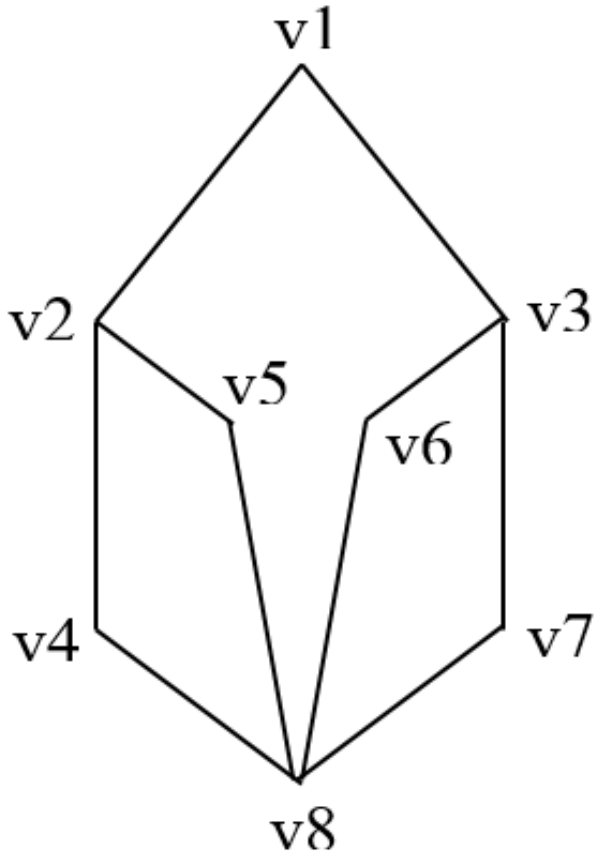
visited

A	B	C	D	E	F	G
y	y	y	y	y	y	y

مثال باستخدام DFS



مثـــال



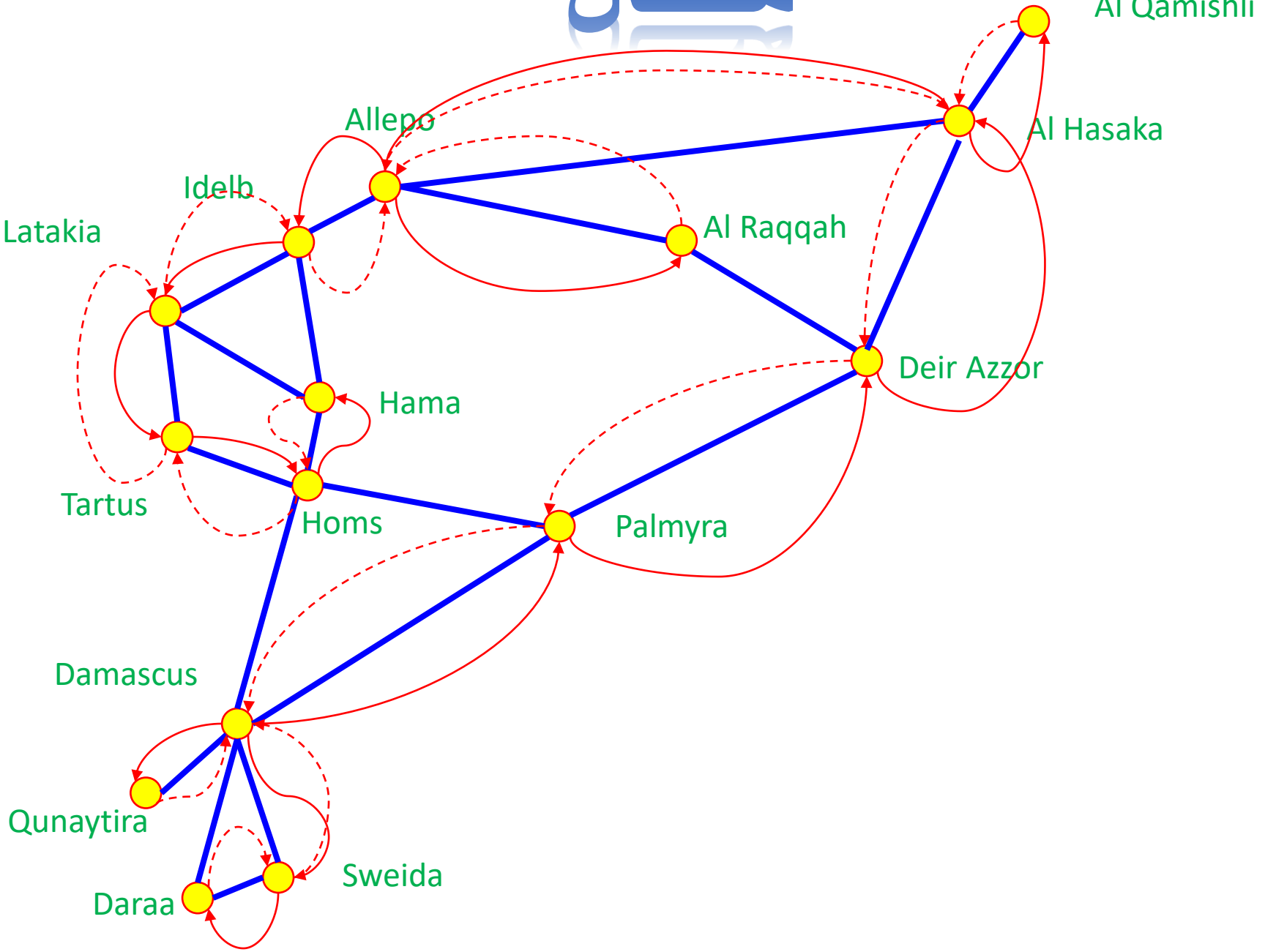
- ليكن لدينا البيان غير الموجه التالي ولتكن v1عقدة البداية:
  - نتيجة مسح البيان بالعمق أولاً (=التجوال بالعمق) :

v1,v2,v4,v8,v5,v6,v3,v7

  - نتيجة مسح البيان بالعرض أولاً (=التجوال بالعرض) :

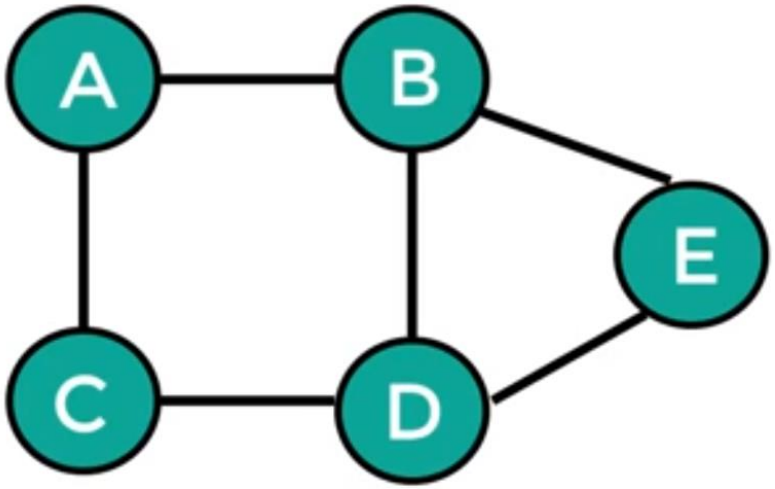
v1,v2,v3,v4,v5,v6,v7,v8

مثـــال

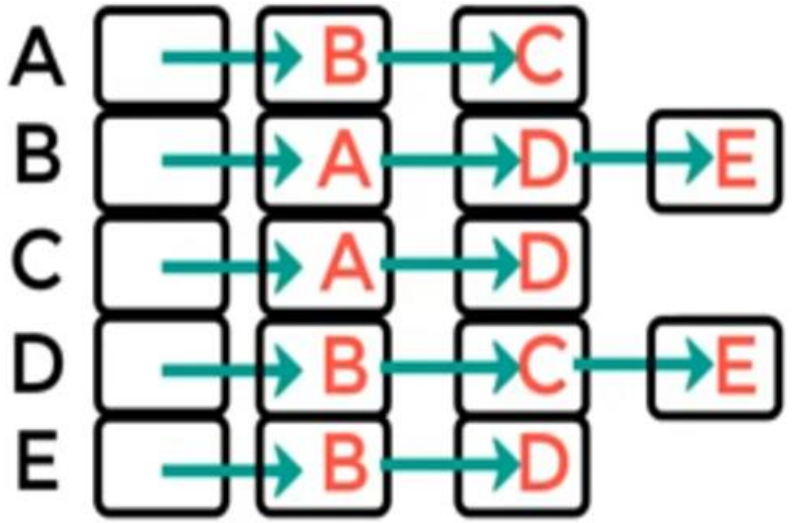

Al Qamishli
Allepo
Al Hasaka
Idelb
Latakia
Al Raqqah
Deir Azzor
Hama
Tartus
Homs
Palmyra
Damascus
Qunaytira
Sweida
Daraa

تحليل الـ DFS

- من أجل البيان G الذي من يملك |V(G)|=n ,e= |E(G)|. يعتمد تعقيد الـ DFS فيه على طريقة التخزين:
  - إذا كان البيان ممثلاً بمصفوفة التجاور يكون التعقيد هو $O(n^2)$.
    - ✓ تحديد الرؤوس المتجاورة مع الرأس v يتطلب زمناً قدره O(n) و بما أنه يتم زيارة كل الرؤوس على الأكثر لذلك يكون زمن التنفيذ الكلي $O(n^2)$
  - إذا كان البيان ممثلاً بقوائم التجاور يكون التعقيد هو $O(n+e)$ وذلك لاننا سنمر في الخوارزمية على e ضلع على الاكثر و يضاف لها كلفة زيارة كل عقدة من العقد.



	A	B	C	D	E
A	0	1	1	0	0
B	1	0	0	1	1
C	1	0	0	1	0
D	0	1	1	0	1
E	0	1	0	1	0

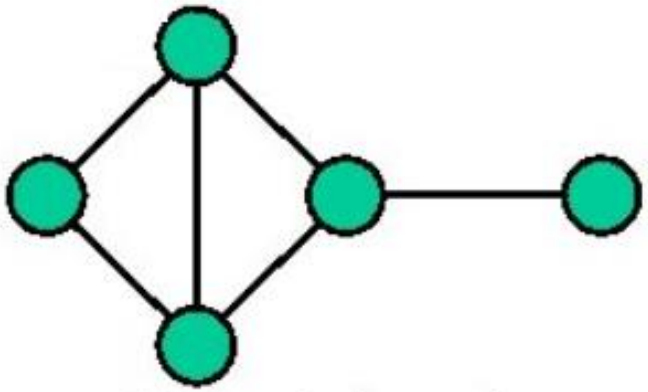


DFS Vs. BFS

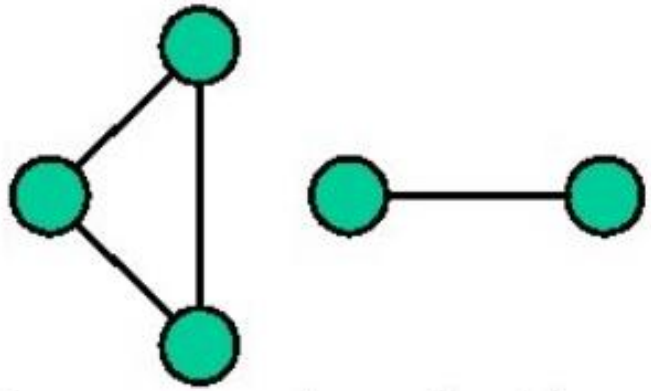
	DFS	BFS
التعقيد	كلاهما يملكان نفس التعقيد	
أقصر طريق	تحتاج إلى تعديل المصفوفة التي تمثل البيان من أجل إيجاد أقل عدد خطوات	أول زيار ة هي الأقصر حكما، لذلك ال BFS هو الحل الأفضل إذا لم يكن لدينا أوزان
استهلاك الذاكرة	الذي يستهلك الذاكرة هو ال Stack	الذي يستهلك الذاكرة هو ال Oueue
ملاحظة	ال Stack المستخدم في ال DFS يخزن في Stack العودية حيث حجم الذاكرة يكون محدود أساسا، أي له حد امتلاء يقوم عند تجاوز ه بالتفريغ على عكس ال Queueالذى يخزن بال Heap (RAM) بالتالى له حجم ذاكرة أكبر	

تطبيقات التجوال – البيان المتصل

- تساعد خوارزمية DFS أو BFS في تحديد ما إذا كان **البيان متصلاً أم لا** : بحيث إذا تمت زيارة **جميع** رؤوس البيان عندئذ يكون البيان مترابطاً وفي غير ذلك يكون البيان غير مترابط .
  - أيضاً، يمكننا إيجاد عدد المركبات المترابطة في بيان غير مترابط .



Connected graph



Non connected graph with two connected components

تطبيقات التجوال- أشجار الامتداد

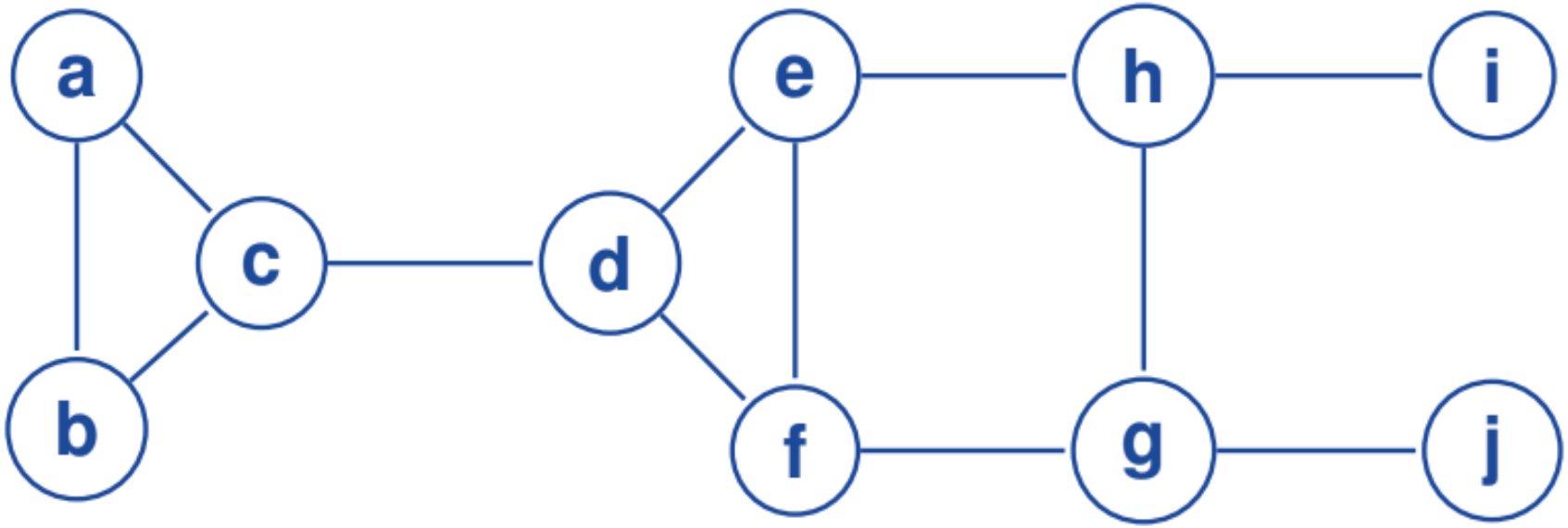
- عند تطبيق إحدى خوارزميتي البحث (بالعمق أولاً أو بالعرض أولاً) على أي رأس من رؤوس **البيان المتصل**، ستتم زيارة جميع العقد وينتج لدينا مجموعتين من الاضلاع:
  1. مجموعة اضلاع شجرية (Tree edges): مجموعة الأضلاع **المستخدمة** في عملية التجوال.
  2. مجموعة اضلاع غير شجرية (Non-Tree edges): مجموعة الأضلاع **المتبقية** .

- نسمي البنية المكونة من **الأضلاع الشجرية** مع **كل رؤوس البيان بشجرة الامتداد (spanning tree)**. ← تمثل **البيان الجزئي الاصغري المتصل** المكون من **n-1** ضلع.
  - بالتالي كل بيان مترابط مكون من n رأساً و n-1 ضلعاً يكون شجرة

- يرمز لشجرة الامتداد الناتجة عن تطبيق خوارزمية البحث
  - بالعمق أولاً بـ depth first spanning tree.
  - بالعرض أولاً بـ breadth first spanning tree .

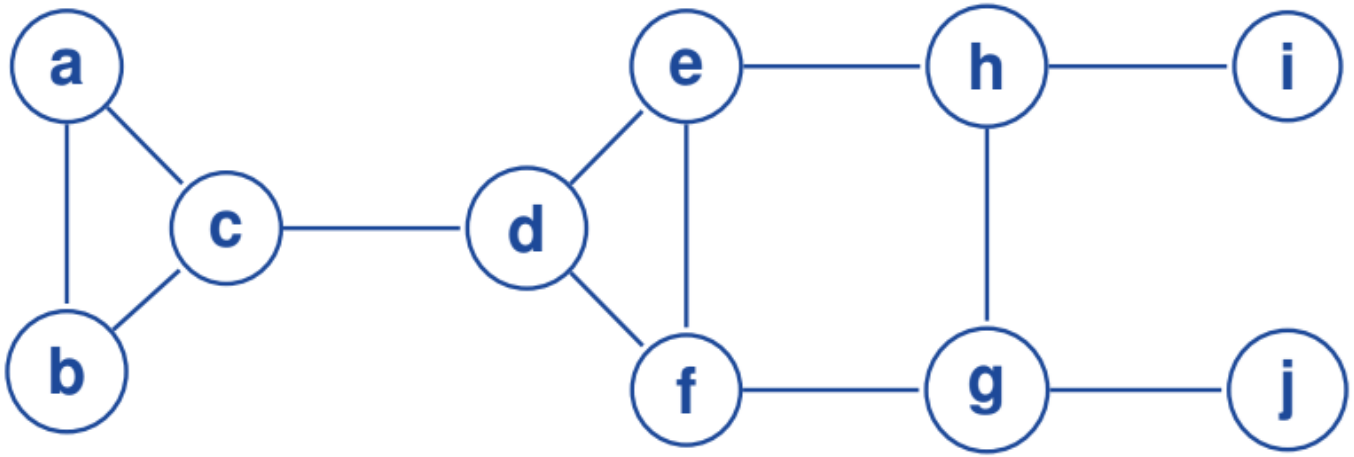
تطبيقات التجوال– أشجار الامتداد

- **مثال1:** قم بإيجاد شجرتي الامتداد الناتجتين عن تطبيق خوارزميتي البحث بالعمق أولاً و بالعرض أولاً على البيان التالي، مع الاخذ بعين الاعتبار ما يلي:
  - عقدة البداية (a).
  - العقد مرتبة ابجدياً.

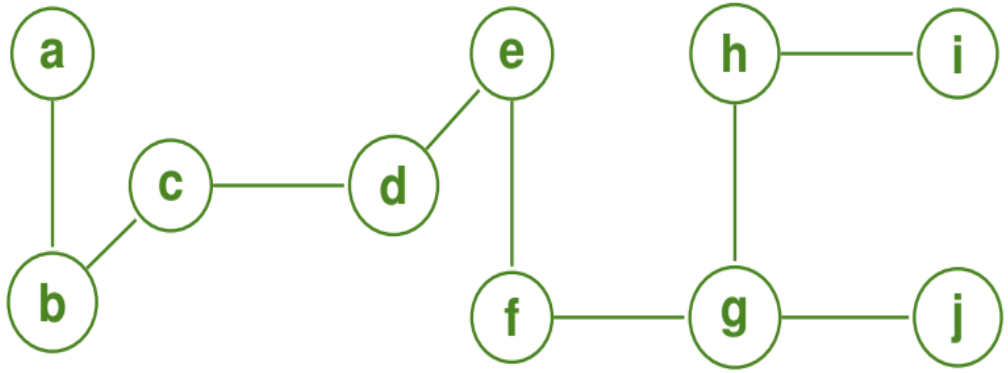


تطبيقات التجوال– أشجار الامتداد

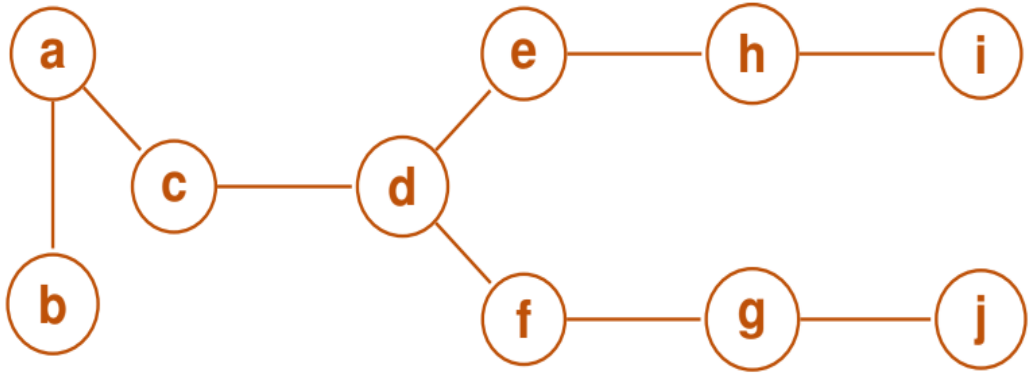
- مثال1:



The spanning tree by DFS:

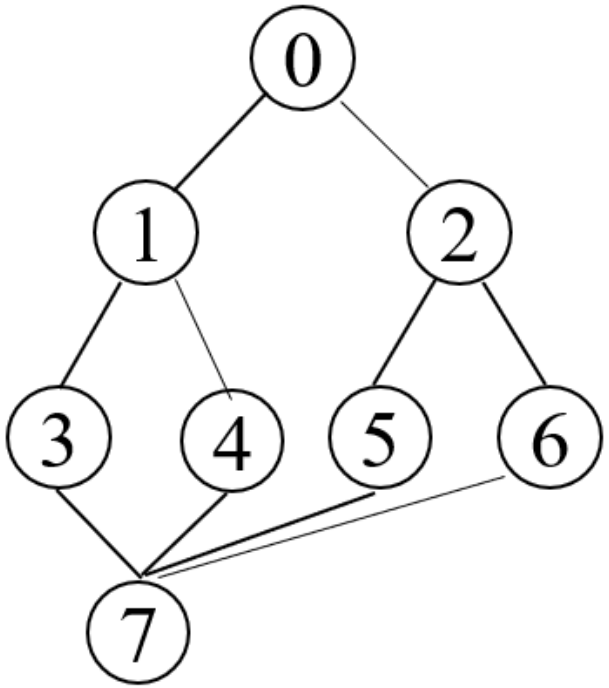


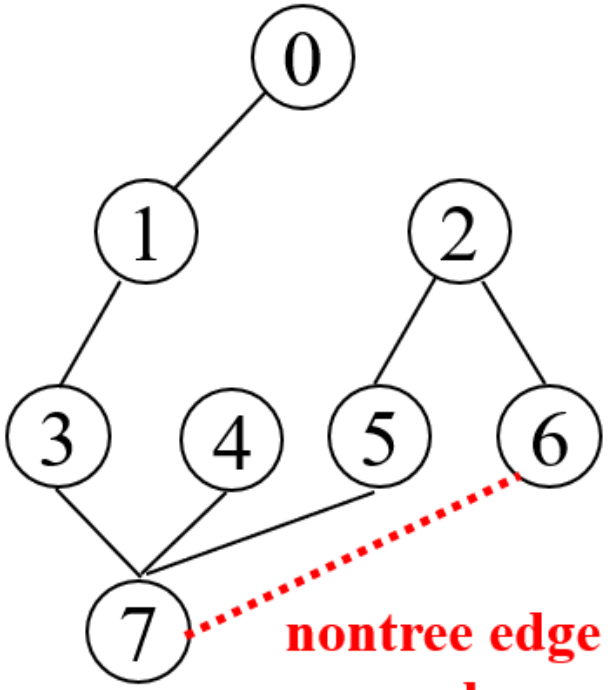
The spanning tree by BFS:



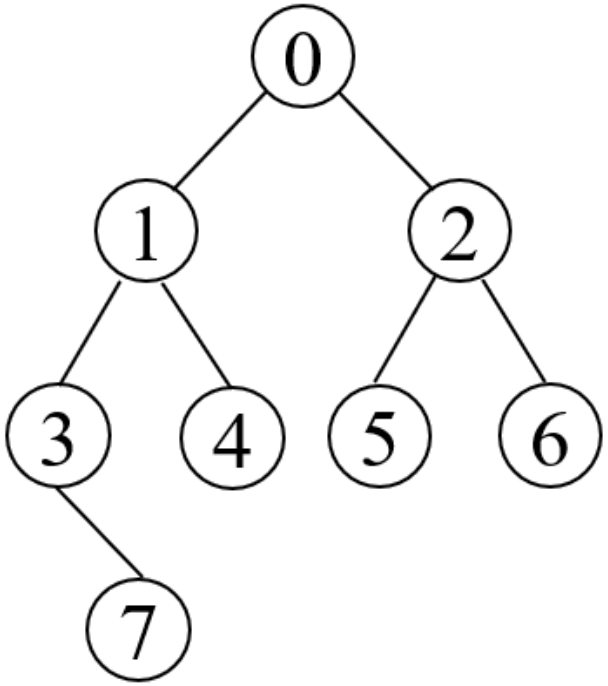
تطبيقات التجوال– أشجار الامتداد

- **ملاحظة:** اضافة **ضلع غير شجري** إلى شجرة الامتداد سوف يؤدي إلى حلقة (cycle).





DFS Spanning



BFS Spanning